Software Requirement Specification (SRS)

Project Title: Question Answering Web Application using BERT

1. Introduction

Purpose:

This project aims to develop a web-based Question Answering (QA) system that allows users to input a paragraph (context) and ask questions related to it.

The system uses a fine-tuned BERT model to extract accurate answers from the provided context.

Scope:

- Accept context paragraph from user.

- Accept question from user.

- Predict answer span using fine-tuned BERT model.

- Display extracted answer on web page.

- Deploy system using Flask and Render cloud.

2. Overall Description

Product Perspective:

The system consists of:

- Frontend (HTML, CSS)

- Backend (Flask API)

- Machine Learning Model (Fine-tuned BERT QA)

- Deployment on Render

System Architecture:

User → Web Interface → Flask Backend → BERT Model → Answer → Display

3. Functional Requirements

FR1: System shall allow users to enter context paragraph.

FR2: System shall allow users to enter a question.

FR3: System shall predict answer using BERT QA model.

FR4: System shall display extracted answer and optional confidence score.

FR5: System shall validate empty inputs and show error messages.

FR6: System shall be deployable on Render using Gunicorn.

4. Non-Functional Requirements

Performance:

- Response time should be within 3–5 seconds.

Usability:

- Simple and responsive UI.

Security:

- Input validation and safe handling of user data.

Scalability:

- Support at least 50–100 concurrent users.

5. Software Requirements

- Python 3.x
- Flask
- Transformers (HuggingFace)
- PyTorch
- Gunicorn
- Render Cloud Platform

6. Dataset Requirements

Model fine-tuned using SQuAD or similar extractive QA dataset.

Dataset structure:

Context | Question | Answer

7. Constraints

- Large BERT model size (~400MB)
- No GPU on free deployment tier
- Possible cold start delay

8. Future Enhancements

- Conversational QA
- Multi-document QA
- Voice-based QA
- User authentication